Amendments of the Claims

The following listing of claims (if entered) will replace all prior versions, and listings, of claims in the above-identified patent application.

Listing of the Claims:

1. (currently amended) A memory controller, comprising:

at least one bus interface, each bus interface being for connection to at least one respective device for receiving memory access requests;

a memory interface, for connection to a memory device over a memory bus;

a plurality of buffers in the memory interface, each of the plurality of buffers sized to store a data burst for a memory access request, each of the plurality of buffers further including a plurality of sub-buffers, each sized to store a data beat of the data burst stored in one of the corresponding plurality of buffers; and

control logic, for placing received memory access requests into a queue of memory access requests,

wherein, in response to a received memory access request requiring multiple data bursts over the memory bus, each of said multiple data bursts is assigned by the control logic to a respective buffer of the plurality of buffers in the memory interface, and data from each of said multiple data bursts is stored by the memory interface in the respective buffer,

wherein, for a wrapping memory access request requiring multiple buffers of the plurality of buffers, data required for each of a beginning and an end of the wrapping memory access request are assigned to respective sub-buffers of a single respective buffer by the control logic, [[the]] a beginning data and an end data for the wrapping memory access request being stored concurrently from a single data burst in the respective sub-buffers of the single respective buffer by the memory interface, the storing of the beginning and end data in the single respective buffer avoiding the need for an additional data burst to obtain the end data, the data required for the end of the wrapping memory access request being cached in one or more of the respective sub-buffers until needed for transfer in response to the wrapping memory access request, and

wherein the control logic records a value of a pointer indicating a first sub-buffer of the single respective buffer storing the end data, such that the control logic is able to return to the indicated first sub-buffer to retrieve the end data from the single respective buffer, and

wherein when accessing the single respective buffer comprising a first part and a second part to return data to the respective device from which a wrapping memory read request requiring multiple data bursts over the memory bus was received, the beginning data is read out from the first part of the single respective buffer, the second part of the single respective buffer is skipped to read out subsequent data from at least one other of said multiple buffers, and the multiple buffers are wrapped around to read out the end data from the second part of the single respective buffer.

2-4. (cancelled)

5. (original) A memory controller as claimed in claim 1, wherein the control logic determines whether a received read access request is a wrapping request which requires multiple memory bursts, and, if so, the control logic allocates each of said memory bursts to a respective one of said buffers.

6. (original) A memory controller as claimed in claim 1, wherein the memory controller is a SDRAM controller,

and said memory interface is suitable for connection to a SDRAM memory device over said memory bus.

7. (previously presented) In a memory controller including at least one bus interface for connection to at least one respective device for receiving memory access requests, a memory interface for connection to a memory device over a memory bus, a plurality of buffers in the memory interface, and control logic for placing received memory access requests into a queue of memory access requests, a method of retrieving data comprising:

in response to a received memory access request requiring multiple data bursts over the memory bus, assigning each of the multiple data bursts to a respective buffer in the plurality of buffers in the memory interface, each of the plurality of buffers being sized to store a data burst for the memory access request, each of the plurality of buffers further including a plurality of sub-buffers, each sized to store a data beat of the data burst stored in one of the corresponding plurality of buffers;

storing data from each of said multiple data bursts in the respective buffer in the memory interface;

for a wrapping memory access request requiring multiple buffers of the plurality of buffers, assigning data required for a beginning and an end of the wrapping memory access request to respective sub-buffers of a single respective buffer to be stored concurrently from a single data burst in the respective sub-buffers of the single respective buffer in the memory interface, the storing of a beginning data and an end data in the single respective buffer avoiding the need for an additional data burst to obtain the end data, the data required for the end of the wrapping memory access request being cached in one or more of the respective sub-buffers until needed for transfer in response to the wrapping memory access request;

recording a value of a pointer indicating a first sub-buffer of the single respective buffer storing the end data; and

using the pointer to return to the indicated first sub-buffer to retrieve the end data,

wherein when accessing the single respective buffer comprising a first part and a second part to return data to the respective device from which a wrapping memory access request requiring multiple data bursts over the memory bus was received, the beginning data is read out from the first part of the single respective buffer, the end data is not read out from the second part of the single respective buffer, then data is read out from at least one other of said buffers, and then the multiple buffers are wrapped around and the end data is read out from the second part of the single respective buffer.

8-10. (cancelled)

11. (previously presented) A method as claimed in claim 7, further comprising determining whether a received read access request is a wrapping request which requires multiple memory bursts, and, if so, performing the step of assigning each of said memory bursts to a respective one of said buffers.

12. (original) A method as claimed in claim 7, wherein the memory controller is a SDRAM controller, and said memory interface receives data from a SDRAM memory device over said memory bus in SDRAM bursts.

13. (previously presented) A programmable logic device, wherein the programmable logic device includes a memory controller, comprising:

at least one bus interface, each bus interface being for connection to at least one respective device formed within the programmable logic device for receiving memory access requests;

a memory interface, for connection to an external memory device over a memory bus;

a plurality of buffers in the memory interface, each of the plurality of buffers sized to store a data burst for a memory access request, each of the plurality of buffers further including a plurality of sub-buffers, each sized to store a data beat of the data burst stored in one of the corresponding plurality of buffers; and

control logic, for placing received memory access requests into a queue of memory access requests,

wherein, in response to a received memory access request requiring multiple data bursts over the memory bus, each of said multiple data bursts is assigned by the control logic to a respective buffer of the plurality of buffers in the memory interface, and data from each of said multiple data bursts is stored by the memory interface in the respective buffer,

wherein, for a wrapping memory access request requiring multiple buffers of the plurality of buffers, data required for each of a beginning and an end of the wrapping memory access request are assigned to respective sub-buffers of a single respective buffer by the control logic, a beginning data and an end data for the wrapping memory access request being stored concurrently from a single data burst in the respective sub-buffers by the memory interface, the storing of the beginning and end data in the single respective buffer avoiding the need for an additional data burst to obtain the end data, the data required for the end of the wrapping memory request being cached in one or more of the respective sub-buffers until needed for transfer in response to the wrapping memory access request; and

wherein the control logic records a value of a pointer indicating a first sub-buffer of the single respective buffer storing the end data, such that the control logic is able to return to the indicated first sub-buffer to retrieve the end data from the single buffer,

wherein when accessing the single respective buffer comprising a first part and a second part to return data to the respective device from which a wrapping memory read request requiring multiple data bursts over the memory bus was received, the beginning data is read out from the first part of the single respective buffer, the second part of the single respective buffer is skipped to read out subsequent data from at least one other of said multiple buffers, and the multiple buffers are wrapped around to read out the end data from the second part of the single respective buffer.

14-17. (cancelled)

18. (previously presented) A memory controller, comprising:

at least one bus interface, each bus interface being for connection to at least one device for receiving memory access requests;

a memory interface, for connection to a memory device over a memory bus;

a plurality of buffers in the memory interface, each of the plurality of buffers sized to store a data burst for a memory access request; and

control logic, for placing received memory access requests into a queue of memory access requests,

wherein, for a wrapping memory access request requiring multiple buffers of the plurality of buffers, data required for each of a beginning and an end of the wrapping memory access request are assigned to sub-buffers of a single buffer by the control logic, and

wherein the control logic records a value of a pointer indicating a first sub-buffer of the single buffer storing the end data, such that the control logic is able to return to the indicated first sub-buffer to retrieve the end data from the single buffer,

wherein when accessing the single respective buffer comprising a first part and a second part to return data to the respective device from which a wrapping memory access request requiring multiple data bursts over the memory bus was received, the beginning data is read out from the first part of the single respective buffer, the end data is not read out from the second part of the single respective buffer, then data is read out from at least one other of said buffers, and then the multiple buffer are wrapped around and the end data is read out from the second part of the single respective buffer.

19. (cancelled)

20. (previously presented) A memory controller as claimed in claim 18, wherein the control logic determines whether a received read access request is a wrapping request which requires multiple memory bursts, and, if so, the control logic allocates each of the memory bursts to one of the buffers.

21. (previously presented) A memory controller as claimed in claim 18, wherein the memory controller is a SDRAM controller, and the memory interface is suitable for connection to a SDRAM memory device over the memory bus.

22. (previously presented) The memory controller of claim 1 wherein each of the plurality of sub-buffers are sized to store a data beat of the data burst stored in one of the corresponding plurality of buffers.

23. (previously presented) The memory controller of claim 22 wherein the end data required for the wrapping memory access request is cached in one or more of the

respective sub-buffers until needed for transfer in response to the wrapping memory access request.

24. (previously presented) The method of claim 7 wherein each of the plurality of sub-buffers are sized to store a data beat of the data burst stored in one of the corresponding plurality of buffers.

25. (previously presented) The method of claim 24 wherein the end data required for the wrapping memory access request is cached in one or more of the respective sub-buffers until needed for transfer in response to the wrapping memory access request.

26. (previously presented) The programmable logic device of claim 13 wherein each of the plurality of sub-buffers are sized to store a data beat of the data burst stored in one of the corresponding plurality of buffers.

27. (previously presented) The programmable logic device of claim 26 wherein the end data required for the wrapping memory access request is cached in one or more of the respective sub-buffers until needed for transfer in response to the wrapping memory access request.